IN THE CLAIMS:

Please amend claims 29 and 37 as shown in this complete set of all pending claims:

1-28. (Canceled)

29. (Currently Amended) A method for accessing a file in a file system in a protected area comprised in secondary storage of a digital processing system comprising a secure random access memory (RAM), the method comprising:

opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of the secure RAM, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in a shared data portion of the secure RAM, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file; and

accessing the file using a file access operation comprised in a file data processing module loaded in the shared execution portion, wherein the data processing module overlays at least a portion of the metadata processing module, and wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer.

30. (Previously Presented) The method of claim 1, wherein the file access operation comprises:

    computing, based on a start index of the portion of data and the cluster size, an index into the buffer of a location of a cluster identifier of a cluster comprising a start of the data;

    using the index to retrieve the cluster identifier from the buffer;

    computing an offset within the cluster of the start of the data; and

    issuing commands to access the data in the cluster starting at the offset.

31. (Previously Presented) The method of claim 1, wherein the cluster identifiers are stored sequentially in the buffer in cluster allocation order.

32. (Previously Presented) The method of claim 1, wherein the sequence of clusters consists of all clusters allocated to the file.

33. (Previously Presented) The method of claim 1, wherein opening the file and accessing the file are preformed in a secure mode of the digital processing system.

34. (Previously Presented) The method of claim 1, wherein each file in the file system has a same number of clusters and the buffer is of a size to store a cluster identifier for all clusters in a file.

35. (Previously Presented) The method of claim 1, wherein the buffer is overwritten each time a file in the file system is opened.

36. (Previously Presented) The method of claim 1, wherein the secondary storage is a secure digital card.

37. (Currently Amended) A machine readable non-volatile storage medium comprising executable instructions that, when executed by a processor of a digital processing system, cause performance of a method for accessing a file in a file system in a protected area comprised in secondary storage of the digital processing system, the method comprising:

opening the file using a file open operation comprised in a file metadata processing module loaded in a shared execution portion of a secure random access memory (RAM) comprised in the digital processing system, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in a shared data portion of the secure RAM, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file; and

accessing the file using a file access operation comprised in a file data processing module loaded in the shared execution portion, wherein the data processing module overlays at least a portion of the metadata processing module, and wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer.

38. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the file access operation comprises:

computing, based on a start index of the portion of data and the cluster size, an index into the buffer of a location of a cluster identifier of a cluster comprising a start of the data;

using the index to retrieve the cluster identifier from the buffer;

computing an offset within the cluster of the start of the data; and

issuing commands to access the data in the cluster starting at the offset.

39. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the cluster identifiers are stored sequentially in the buffer in cluster allocation order.

40. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the sequence of clusters consists of all clusters allocated to the file.

41. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein opening the file and accessing the file are preformed in a secure mode of the digital processing system.

42. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein each file in the file system has a same number of clusters and the buffer is of a size to store a cluster identifier for all clusters in a file.

43. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the buffer is overwritten each time a file in the file system is opened.

44. (Previously Presented) The machine readable non-volatile storage medium of claim 37, wherein the secondary storage is a secure digital card.

45. (Previously Presented) A digital processing system comprising:

a first secondary storage comprising a file system in a protected area, wherein the file system comprises a plurality of files;

a second secondary storage comprising a file metadata processing module comprising a file open operation and a file data processing module comprising a file access operation; and

a secure random access memory comprising a shared execution memory portion and a shared data memory portion,

wherein to open a file in the plurality of files,

the file metadata processing module is loaded in to the shared execution memory portion, and

the file open operation is executed, wherein the file open operation traverses a file access table (FAT) of the file system to determine a sequence of clusters allocated to the file and stores a cluster identifier for each cluster in the sequence in a buffer comprised in the shared data portion, wherein the cluster identifiers are stored in the buffer such that each cluster identifier is locatable by an index computed using a cluster size and a start offset of data in the file, and

wherein to access the file,

the file data processing module is loaded in the shared execution memory portion, wherein the data processing module overlays at least a portion of the metadata processing module, and

the file access operation is executed, wherein the file access operation accesses a portion of data in the file using at least one cluster identifier stored in the buffer.

46. (Previously Presented) The digital processing system of claim 45, wherein the file access operation:

      computes, based on a start index of the portion of data and the cluster size, an index into the buffer of a location of a cluster identifier of a cluster comprising a start of the data;

      uses the index to retrieve the cluster identifier from the buffer;

      computes an offset within the cluster of the start of the data; and

      issues commands to access the data in the cluster starting at the offset.

47. (Previously Presented) The digital processing system of claim 45, wherein the cluster identifiers are stored sequentially in the buffer in cluster allocation order.

48. (Previously Presented) The digital processing system of claim 45, wherein the sequence of clusters consists of all clusters allocated to the file.

49. (Previously Presented) The digital processing system of claim 45, wherein the file open operation and the file access operation are preformed in a secure mode of the digital processing system.

50. (Previously Presented) The digital processing system of claim 45, wherein each file in the file system has a same number of clusters and the buffer is of a size to store a cluster identifier for all clusters in a file.

51. (Previously Presented) The digital processing system of claim 45, wherein the buffer is overwritten each time a file in the file system is opened.

52. (Previously Presented) The digital processing system of claim 45, wherein the first secondary storage is a secure digital card.